

はじめよう/

ジブン専用パソコン

ゲーム

工作

プログラミング



ラズベリーパイ Raspberry Pi 大活用

第12回 キッツキロボを改造しよう

今回は、前回つくったキッツキロボをグレードアップさせるゾ。首を前後だけでなく、左右にも動かせるようにして、どんな質問にも答えてくれる占い機能を追加しちゃおう!

監修・原案 / 青山学院大学客員教授 阿部和広
構成・文 / 塩野祐樹

キットの情報は
KoKa Shop!へ
定期購読者割引
特典あり
購入ページ



Scratchベースで動かそう!
Studuinoでラクラク電子工作

prog.kodomonokagaku.com/studuino.html

ラズパイがなくても、
普通のパソコンで遊ぶ
ことができるよ

キッツキロボと プログラミング環境の準備

まずは、前回つくったキッツキロボを改造しよう。いったんキッツキロボを分解してDCモーターを取り付けるんだ。前回のキッツキロボのつくり方と改造の仕方は、「スタプロ」の中にある「お手軽パソコン ラズベリーパイ大活用」を見てね。

prog.kodomonokagaku.com/jibun/

これが
キッツキロボ
改造版だ

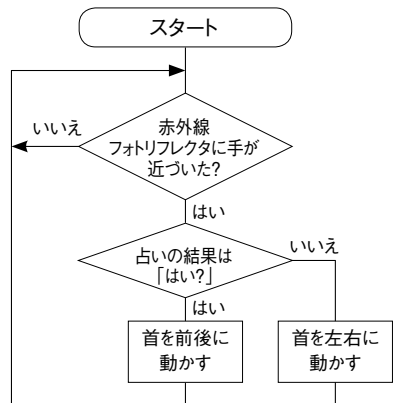


また、ブロックプログラミング環境 (BPE) の起動と、入出力の設定もしておこう。これも前回と同じだよ。右ページ上で簡単に復習しているけど、詳しくは前回の記事を見てね。

プログラムの流れ

今回のプログラムでは、首を前後に振る動きと左右に振る動きがあるので、それぞれの動きに対応したプログラムを「関数」にして呼び出しているよ。「関数」とは、決まった処理を行うプログラムのかたまりのことだよ。

プログラム全体の流れはこんな感じだ。赤外線フォトフレクタに手を近づけると、占いが



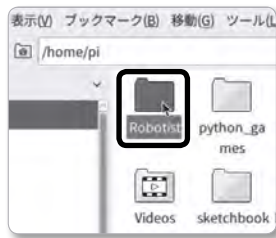
BPEの起動と入出力設定

BPEの起動

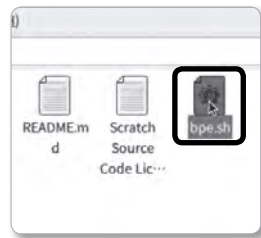
①画面左上にあるアイコンから、ファイルマネージャを開く。



②「/home/pi」にある「Robotist」フォルダを開く。



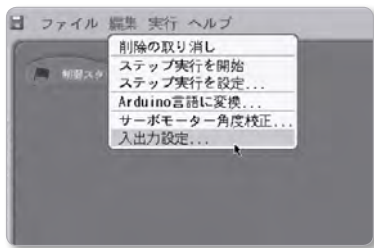
③「bpe.sh」をダブルクリックすると、BPEが起動する。



「ファイルを実行する」ダイアログが表示されたら、「実行」ボタンをクリックしよう。

入出力の設定

①「編集」メニューから「入出力設定...」を開く。



②「A7」の「音センサー」を「赤外線フォトリフレクタ」に変更する。サーボモーターの「D9」と、DCモーターの「M1」にチェックが入っていることを確認する。



スタート。「はい」と「いいえ」をランダムに選び、「はい」のときは首を前後に、「いいえ」のときは首を左右に振るようにしよう。

キツツキの首を前後に動かす

それでは、「実行」メニューの「テストモード開始」でテストモードに切り替えてから、プログラミングを始めよう。

注意!

テストモードになっている間は、絶対にミニUSBケーブルを抜かないようにしましょう。通信が切れると、動かなくなったり、再起動が必要になったりするよ。

まずは、キツツキの首を前後に動かすプログラムだ。これには「動き」カテゴリの「サーボモーター D9を90度にする」を使おう。

さっき説明した「関数」が、ここで登場しているね。このように「hai ▼関数をコールする」ブロックから、「hai ▼関数」の下に書いたプログラムを呼び出すことができるんだ。

新しい関数をつくるときは、「▼関数」か「▼関数をコールする」の「▼」をクリックして「新規...」を選び、キーボードから名前を入力しよう。使える文字は、アルファベットと数字だけだ。ここでは「はい」の動きなので「hai」という名前にしたよ。



電池ボックスのスイッチをONにしてから、緑の旗をクリックすると、頭が垂直に立ち上がってから1秒後に0.3秒間、45°のおじぎをするよ。

注意!

電池ボックスのスイッチがONになっているときは、絶対に手で首を動かさないようにしましょう。サーボモーターが壊れることがあるよ。

キッツキの首を左右に動かす

キッツキの首を左右に回す、つまり「いいえ」の動きをするには、DCモーターを使うよ。DCモーターは、サーボモーターのように回す角度を正確に決めることができない。その代わり、回転する速さと向きを指定できるんだ。

次のプログラムを入力して緑の旗をクリックしてみよう。「はい」の動きの1秒後に、「いいえ」の動きをしたかな？

The program starts with a 'Control Start' block. It sets 'Servo Motor D9' to 90 degrees, waits 1 second, and calls the 'hai' function. It then waits 1 second and calls the 'iie' function. The 'iie' function block contains: DC Motor M1 speed set to 100, DC Motor M1 reverse, wait 0.2 seconds, DC Motor M1 forward, wait 0.4 seconds, DC Motor M1 reverse, wait 0.2 seconds, and DC Motor M1 stop.

「iie▼関数」では、まずDCモーターが回る速さを「100」にしている。そして、「逆転」の向き（上から見て反時計回り）で0.2秒間回してから、「正転」の向き（上から見て時計回り）で0.4秒間回し、再び「逆転」で0.2秒回している。つまり、向かって右に首を振ってから、左に振り、最後に元の位置に戻るという動きになっている。

もしかしたら、ぴったり同じ位置には戻らず、ちょっとずれているかもしれないね。時間と速さで動かすDCモーターは、電池の残量やモーターの

特性、摩擦といったいろいろな条件や誤差などによって、正確に動かすことができないんだ。このような制御方法を「オープンループ制御」というよ（サーボモーターは「クローズドループ制御」だ）。気になるときは、秒数や速さを微調整して、最後にぴったり正面を向くようにしてみよう。

首を震わせる

「はい」と「いいえ」の動きに加えて、もう1つ、首を震わせる動きもつくっておこう。これは、占いがスタートしてから結果を出すまでの間の動きだ。名前を「buruburu▼関数」として、サーボモーターを前後に10回、小刻みに動かそう。

The program starts with a 'Control Start' block. It sets 'Servo Motor D9' to 90 degrees, waits 1 second, and calls the 'hai' function. It then waits 1 second and calls the 'iie' function. The 'iie' function block contains: 'buruburu' function call, wait 10 loops, Servo Motor D9 set to 85 degrees, wait 0.1 seconds, Servo Motor D9 set to 95 degrees, wait 0.1 seconds, and Servo Motor D9 set to 90 degrees.

ところで、キッツキの動きがなんだかごちないなあと思うことがあるかも。実はテストモードでは、通信に時間がかかるので、正確に動かないことがあるのだ。この後で紹介する「プログラムの転送」を行うとスムーズに動くので、それまでちょっとガマンしてね。

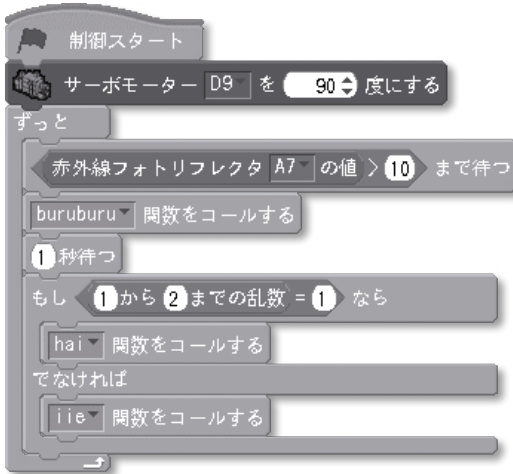
占いのプログラム

これで、動きに必要なプログラムはそろったので、占いの本体となるプログラムにとりかかろう。

赤外線フォトリフレクタに手を近づけると、占いがスタート。占っている間は首がぶるぶる震えて、占い結果の「はい」か「いいえ」は乱数で選ぶことにしよう。

「ずっと」と「～まで待つ」は「制御」カテゴリ、「赤外線フォトリフレクタ▼の値」は「調べる」カテゴリにあるよ。「>」、「1から2までの乱数」、「=」は「演算」カテゴリだ。数字を変えるときは、数字をクリックしてキーボードから半角で入力してね。

「制御スタート」のプログラムを次のように組み替えよう。できたら、緑の旗をクリックして、質問を考えてから手を赤外線フォトリフレクタに近づけてみよう。ちゃんと占ってくれるかな？



では、プログラムを詳しく見ていこう。

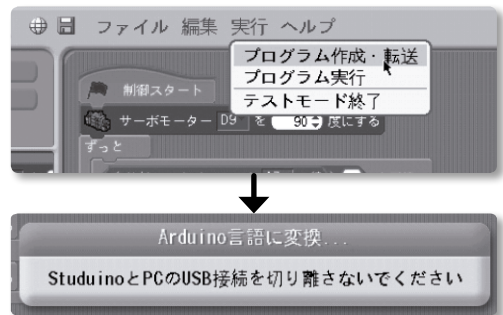
緑の旗をクリックされると、まず、サーボモーターの角度を90°にして、首を垂直に立てる。そして、赤外線フォトリフレクタの値が10より大きくなるまで、つまり手が近づくまで待っている。手が近づいたら首を震わせて、1秒待ってから乱数で1か2の値を取得する。値が1だったら「はい」の動き、そうでなかったら「いいえ」の動きをするよ。その後は、再び手が近づくのを待つんだ。

乱数の範囲を変えて確率を変化させたり、首の動きを追加したりしてみてもおもしろいよ。

プログラムの転送

ここまでは、スタディーノとラズパイをミニUSBケーブルでつなぎ、テストモードで動かしていたね。プログラムが完成したら、スタディーノにプログラムを転送することで、ラズパイと切り離しても動かすことができるんだ。

「実行」メニューから「プログラム作成・転送」を選ぶと、プログラムがスタディーノに転送されるぞ。



「StuduinoとPCのUSB接続を切り離さないでください」という表示が消えたら、転送は終了だ。ミニUSBケーブルを抜いて、スタディーノ単独でもちゃんと動くことを確認しよう。このとき、電池ボックスのスイッチをONにするのを忘れないうね。スタディーノだけで動かすときは、本体の丸い「RESET」ボタンが緑の旗の代わりになるよ。

保存と片付け

つくったプログラムを保存するには、「ファイル」メニューの「名前を付けて保存...」を選ぶ。これはスクラッチといっしょだね。

また、テストモードのときは、ミニUSBケーブルを抜く前に「実行」メニューの「テストモード終了」を選ぶことを忘れないでね。BPEの終了は、ウィンドウ右上の「×」をクリックだ。電池ボックスのスイッチをOFFにしたら、つくったロボットを分解しても大丈夫だよ。

次回もラズパイとスタディーノを連携させたプログラムを紹介する予定だよ。どんなロボットをつくるのか、お楽しみに～。